|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *B7* | | |
| Homework number: | *HW10* | | |
| Due date: | 13/12/2022 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| 1 *Giampà Simone* |  |  | *x* |
| 2 *Massa Giacomo* | *x* |  |  |
| 3 *Raduzzi Lucafrancesco* | *x* |  |  |
| 4 *Micelli Johanna* |  |  | *x* |
| 5 *Galimberti Claudio* |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | HW10a\_Keyboard | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| SETUP: We enabled PC2, PC3, PC12, PC13 pins as GPIO\_Input and PC8, PC9, PC10, PC11 pins as GPIO\_Output.  CODE: The columns of the buttons are read iteratively, by activating one column at a time. The function HAL\_Delay(50) is used to wait for the enabling of the column so that the state of the buttons can be read, to find the button that has been pressed. The code also makes sure that a button press is shown only once, and not more than one time if the button is kept pressed for a long time. In this version, the button presses are shown for every button press, regardless of the duration of the press. | | | |
| Professor comments: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | HW10b\_Keyboard\_Timers | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| SETUP: We enabled PC2, PC3, PC12, PC13 pins as GPIO\_Input and PC8, PC9, PC10, PC11 pins as GPIO\_Output. We configured TIM2 to get a timer that generates an interrupt every 50 milliseconds and TIM5 to have a timer that serves as a stopwatch, counting time until 10 seconds elapse.  CODE: Every 50 milliseconds the interrupt expires enabling a different column at each iteration, looping on all 4 columns. After a column is enabled, all the rows states are read to find which button has been pressed in that column. Then the timer TIM5 starts, counting the time during which a button is pressed. This time is represented as the counter value obtained directly from the timer.  A button press is printed on the emulated terminal if the button is kept pressed until the time threshold is reached (this time threshold can be controlled by changing the variable time\_button\_press). This way a button press is shown only if the button is being pressed for the defined time threshold or longer.  The TIM5 counter must be reset to 0 with \_\_HAL\_TIM\_SetCounter(&htim5, 0) every time the button is released or the time threshold is reached. This is done because the timer counts until 10 seconds, and since the time needed to keep the key pressed is less than 10 seconds, this requires starting the counting in the next iteration from zero. Also, stopping TIM5 does not reset the counter value, so a manual reset is required.  The code also checks whether a key has been pressed for enough time (longer than the defined threshold) and whether the single button press has not already been shown on the terminal. | | | |
| Professor comments: | | | |